We Claim:

1. In an electronic device, a method for maintaining a timed-event list of operations to be performed by said electronic device, said method, comprising the steps of:

5   providing a first data structure, said data structure of a size larger than the expected number of events occurring in said electronic device at any one time,

providing a plurality of second data structures referenced by said first data structure, said second data structures holding third structures, each said third structure encapsulating data about a timed event, said timed events to be performed by said

10   electronic device,

inserting a third structure into a selected one of said second data structures referenced by said first data structure, said selected one of said second data structures selected based on the time for execution of a timed event encapsulated by said third structure, the size of said first data structure and a current time.

15

2. The method of claim 1 comprising the further step of:

serially traversing said first data structure at periodic timed intervals.

3. The method of claim 2 comprising the further steps of:

20   serially traversing a second data structure referenced by said first data structure in the time period between said periodic timed intervals; and

inspecting a third structure held by said second data structure.

4. The method of claim 3 comprising the additional steps of:

25   retrieving said inspected third structure from said second data structure; and

executing a timed event encapsulated by said third structure.

5. The method of claim 3 comprising the further steps of:

removing said inspected third structure from said second data structure; and

30   inserting said inspected third structure into a different second data structure referenced by said first data structure based on the time for execution of a timed event encapsulated by said third structure, the size of said first data structure and a current time.

6. The method of claim 5 comprising the further steps of:
    retrieving said inspected third structure from said second data structure; and
    executing a timed event encapsulated by said third structure.

7. The method of claim 3 wherein said first data structure is an array.

8. The method of claim 3 wherein said plurality of said second data structures are linked lists.

9. The method of claim 3 wherein said plurality of said second data structure are doubly-linked lists.

10. The method of claim 3 wherein said third structure encapsulates data about a timed event that is part of a computer simulation.

11. The method of claim 1 wherein said electronic device is a real-time computer system.

12. The method of claim 1 wherein said electronic device is a network switch.

13. The method of claim 1 wherein a hashing algorithm is used to select the insertion point of said third structure.

14. In a computer system, a method for maintaining a timed-event list, said method, comprising the steps of:
    providing an array of memory locations, said memory locations numbering more than the expected number of events occurring in said computer system at any one time,
    providing a plurality of linked lists referenced by said memory locations in said array, said linked lists including nodes, each said node encapsulating data about a timed event to be performed by said computer system;
    inserting a node into a selected one of said linked lists referenced by said memory locations in said array, said selected one of said linked lists selected based on the time for execution of a timed event encapsulated in said node, the size of said array and a current time.

11

15. The method of claim 14 wherein said linked lists are doubly-linked lists.

16. The method of claim 14 comprising the further step of:

     serially traversing said array at periodic timed intervals.

17. The method of claim 16 comprising the further steps of:

     serially traversing a linked list referenced by said memory locations in said array; and

     inspecting a node in said linked list encapsulating data about a timed event.

18. The method of claim 17 comprising the further steps of:

     retrieving said inspected node from said linked list; and

     executing a timed event encapsulated by said inspected node.

19. The method of claim 17 comprising the further steps of:

     removing said inspected node from said linked list; and

     inserting said inspected node into into a different linked list referenced by a different memory location in said array based on the time for execution of a timed event encapsulated by said inspected node, the size of said first data structure and a current time.

20. The method of claim 19 comprising the further steps of:

     retrieving said inspected node from said different linked list; and

executing a timed event encapsulated by said inspected node.

21. The method of claim 14 wherein said computer system is a real-time system.

22. The method of claim 14 wherein said timed event list holds events for a computer simulation.

23. In a system, a medium holding computer-executable instructions for a method, said method, comprising the steps of:

     providing a first data structure, said data structure of a size larger than the expected number of events occurring in said system at any one time,

providing a plurality of second data structures referenced by said first data structure, said second data structures holding third structures, each said third structure encapsulating data about a timed event, said timed events to be performed by said system,

5   inserting a third structure into a selected one of said second data structures referenced by said first data structure, said selected one of said second data structures selected based on the time for execution of a timed event encapsulated by said third structure, the size of said first data structure and the current time.

10   24. The method of claim 23 comprising the further step of:

serially traversing said first data structure at periodic timed intervals.

25. The method of claim 24 comprising the further steps of:

serially traversing a second data structure referenced by said first data structure

15   in the time period between said periodic timed intervals; and

inspecting a third structure in said second data structure, said third structure encapsulating information about a timed event.

26. The method of claim 25 comprising the additional steps of:

20   retrieving said inspected third structure from said second data structure; and

executing a timed event encapsulated by said third structure.

27. The method of claim 25 comprising the further steps of:

removing said inspected third structure from said second data structure; and

25   inserting said inspected third structure into a different second data structure referenced by said first data structure based on the time for execution of a timed event encapsulated by said third structure, the size of said first data structure and the current time.

30   28. The method of claim 27 comprising the further steps of:

retrieving said third structure from said second data structure; and

executing a timed event encapsulated by said third structure.

29. In an optical network, a switching apparatus comprising:

   a processor;

   an event list, said event list comprised of a first data structure holding references

5 to a plurality of second data structures, said second data structures holding third

   structures, said third structures encapsulating information about events scheduled to be

   executed by said switching apparatus; and

   a management facility for maintaining said event list, said management facility

   inserting third structures into selected second data structures referenced by said first data

10 structures based on the time for execution of a timed event encapsulated by said third

   structure, the size of said first data structure and a current time.

30. The apparatus of claim 29 wherein said first data structure is an array

15 31. The apparatus of claim 29 wherein said second data structure is a linked list.

32. The apparatus of claim 29 wherein said second data structure is a doubly-linked list.